CLAIMS

What is claimed is:

1. A method in a computer graphics system for rendering on an output device, having a finite number of pixels, a non-zero thickness line segment with reduced aliasing, comprising:

expanding an edge of the line segment touching but not covering a pixel center of the line segment to be rendered on the output device so that the expanded line segment covers the center of the pixel touched;

determining, using the pixel centers, the pixels to be included in the expanded line segment, the line segment being distinguishable from a background over which said line segment is rendered by having a shade different from a shade of the background; and

for each pixel that is included in said non-zero thickness expanded line segment,

determining the area of the pixel partially or fully covered by said line segment; and

based on the area of the pixel covered, determining a shading value for the pixel by interpolating between the shade of said line segment and the shade of the background.

2. A method for rendering a non-zero thickness line segment as recited in claim 1,

wherein the output device has a x-y coordinate system established thereon and the pixels of the output device each have centers with an x-y coordinate; and

wherein the step of expanding an edge includes moving the edge of said line segment by an amount equal to (a+b)/2a in the x-direction to include the center of a pixel that has a corner traversed by an edge of said line segment, wherein a is greater than or equal to zero and b is greater than or equal to zero.

3. A method for rendering a non-zero thickness line as recited in claim 1,

wherein the output device has a x-y coordinate system established thereon and the pixels of the output device each have centers with an x-y coordinate;

wherein an edge of said line segment has an equation $ax + by + c = 0$; and

wherein the step of expanding an edge includes altering the equation of an edge of said line segment by adding an amount $(|a|+|b|)/2$ to the c parameter of the equation.

4. A method for rendering a non-zero thickness line as recited in claim 1,

wherein the output device has a x-y coordinate system established thereon and the pixels of the output device each have centers with an x-y coordinate; and

wherein the step of determining the pixels to be included in the non-zero thickness line includes:

evaluating the equation of an edge of said expanded line segment with the x and y-coordinates of the center of each pixel; and

testing whether the result of the computation is greater than or equal to zero.

5. A method for rendering a non-zero thickness line as recited in claim 4, wherein evaluating the equation of an edge of said expanded line segment with the x and y-coordinates of the center of each pixel includes computing $ax_0 + by_0 + c + (|a|+|b|)/2$, where x_0 and y_0 are the coordinates of the pixel center.

6. A method for rendering a non-zero thickness line as recited in claim 1,

wherein the shade of the pixel covered and the shade of the background are each indicated by a numerical value; and

wherein the step of determining the shading value of the pixel by interpolating includes:

forming a first product of the shade numerical value of said line segment and a fraction f representing the area of the pixel covered;

forming a second product of the shade numerical value of the background and a fraction $(1-f)$ representing the area of the pixel not covered; and

adding the first and second products.

7. A method for rendering a non-zero thickness line as recited in claim 1,

wherein the output device has a x-y coordinate system established thereon and said line segment has a slope factor sf related to the slope of said line segment and a parameter p proportional to an x-distance between an edge of said line segment traversing a pixel and a pixel boundary; and

wherein, for an edge of said line segment that traverses a partially covered pixel, the step of determining the area of a partially covered pixel includes:

determining that the area covered is less than or equal to a first predetermined limit; and

computing the triangular area covered by said line segment.

8. A method for rendering a non-zero thickness line as recited in claim 7,

wherein the parameter p is equal to the product of the slope factor sf and the distance between an edge of said line segment traversing the pixel and a pixel boundary;

wherein the line segment has a slope m and the slope factor sf equals $m/(m+1)$; and

wherein the step of computing the triangular area covered by said line segment includes forming a product $\frac{1}{2} * p^2 * (1-sf)^{-1} * sf^{1}$ to find the area.

9. A method for rendering a non-zero thickness line as recited in claim 1,

wherein the output device has an x-y coordinate system established thereon and said line segment has a slope factor sf related to the slope of said line segment and a parameter p proportional to an x-directed distance between an edge of said line segment traversing a pixel and a pixel boundary; and

wherein, for an edge of said line segment that traverses a partially covered pixel, the step of determining the area of a partially covered pixel includes:

determining that the area covered is greater than a first predetermined limit;

computing the maximum triangular area covered by said line segment;

computing the area of a parallelogram covered by said line segment; and

computing the sum of the maximum triangular area and the parallelogram area.

10. A method for rendering a non-zero thickness line as recited in claim 9, wherein the slope factor sf equals $m/(m+1)$, where m is the slope of said line segment.

11. A method for rendering a non-zero thickness line as recited in claim 9, wherein the p parameter equals the product of said x-directed distance and the slope factor sf.

12. A method for rendering a non-zero thickness line as recited in claim 11, wherein said x-directed distance is computed as the quotient $ax_0+by_0 + c + (|a|+|b|)/2$ and a, where x_0 and y_0 are the coordinates of the center of the pixel and the line segment edge has an equation $ax+by+c=0$.

13. A method for rendering a non-zero thickness line as recited in claim 9, wherein the first predetermined limit is the maximum area triangular area covered by said line segment traversing through the pixel.

14. A method for rendering a non-zero thickness line as recited in claim 9, wherein the step of computing the maximum triangular area covered by said line segment includes forming a product $\frac{1}{2} * (1-sf) * sf^{1}$ to find the maximum triangular area.

15. A method for rendering a non-zero thickness line as recited in claim 9, wherein the step of computing the parallelogram area covered by said line segment includes forming a sum of $p*sf^{1}$ and $(1-sf^{1})$ to find the parallelogram area.

16. A method for rendering a non-zero thickness line as recited in claim 1, wherein, for an edge of said line segment that traverses a partially covered pixel, the step of determining the area of a partially covered pixel includes:

determining that the area covered is greater than a second predetermined limit, leaving a triangular area not covered;

computing the triangular area not covered by said line segment; and

computing the difference between one and the triangular area not covered to find the area of the pixel covered.

17. A method for rendering a non-zero thickness line as recited in claim 16, wherein the second predetermined limit is the sum of the maximum triangular area and the maximum parallelogram area of said line segment traversing the pixel.

18. A method for rendering a non-zero thickness line as recited in claim 16, wherein the step of computing the triangular area covered by said line segment includes forming a product $\frac{1}{2} * p^2 * (1\text{-}sf)^{-1} * sf^{1}$ to find the triangular area covered.

19. A method for rendering a non-zero thickness line as recited in claim 1, wherein when two parallel edges of said line segment traverse a partially covered pixel, the step of determining the area of the partially covered pixel includes:

computing a first area of the pixel not covered by the first parallel edge;

computing a second area of the pixel not covered by the second edge; and

summing the first and second areas and subtracting the sum from one.

20. A method for rendering a non-zero thickness line as recited in claim 1, wherein when a first edge along said line segment and a second edge orthogonal to said line segment traverse a partially covered pixel, the step of determining the area of the partially covered pixel includes:

computing a first area of the pixel not covered by the first parallel edge and subtracting the first area from one to form a first difference;

computing a second area of the pixel not covered by the second edge and subtracting the second area from one to form a second difference; and

forming a product of the first and second differences.

21. A method for rendering a non-zero thickness line as recited in claim 1, wherein when two parallel edges and a third orthogonal edge of said line segment traverse a partially covered pixel, the step of determining the area of the partially covered pixel includes:

computing a first area of the pixel not covered by the first parallel edge;

computing a second area of the pixel not covered by the second parallel edge; and

summing the first and second areas and subtracting the sum from one to form a first difference;

computing a third area of the pixel not covered by the third orthogonal edge and subtracting the third area from one to form a second difference; and

forming a product of the first difference and the second difference.

22. A method for rendering a non-zero thickness line as recited in claim 1, wherein, when two parallel edges and a third and forth orthogonal edge of said line segment traverse a partially covered pixel, the step of determining the area of the partially covered pixel includes:

computing a first area of the pixel not covered by the first parallel edge;

computing a second area of the pixel not covered by the second parallel edge; and

summing the first and second areas and subtracting the sum from one to form a first difference;

computing a third area of the pixel not covered by the third orthogonal edge;

computing a fourth area of the pixel not covered by the fourth orthogonal edge;

summing the third and fourth areas and subtracting the sum from one to form a second difference; and

forming a product of the first difference and the second difference.

23. A graphics pipeline for rendering a non-zero thickness line segment on an output device, having a finite number of pixels, comprising:

an interpolator processor for computing a parameter proportional to a displacement representing an x-directed distance between an edge of said line segment and the boundary of a pixel that is partially covered by said line segment, the edge of said line segment having a known edge relation that defines a slope factor for the edge, the pixel having a center with known x and y coordinates; and

a shading processor for computing the area of the pixel covered by said line based on the displacement parameter, the slope factor of said line segment, the edge relation and the x and y coordinates of the pixel center.

24. A graphics pipeline as recited in claim 23, wherein said x-directed distance is computed as the quotient $ax0+by0 + c + (a+b)/2$ and a, where x0 and y0 are the coordinates of the center of the pixel and the edge relation is $ax+by+c \geq 0$, and the slope factor is $a/(a+b)$.

25. A graphics pipeline as recited in claim 23, said displacement parameter is computed as $sf * x0+(1-sf) * y0 + sf * c/a + 1/2$, where x0 and y0 are the coordinates of the center of the pixel, the edge relation is $ax+by+c \geq 0$, and the slope factor is $a/(a+b)$.

26. A graphics pipeline as recited in claim 23,

wherein the pixel has center coordinates $(x0, y0)$, the edge relation of the line segment is $ax+by+c \geq 0$, and the slope factor is $a/(a+b)$;

wherein said interpolator process evaluates interpolator function, $Ps + Px * x + Py * y$, to perform interpolation calculations; and

wherein displacement parameter is computed by setting Ps to $(sf * c/a + \frac{1}{2})$, Px to sf and Py to $(1-sf)$ and evaluating the interpolator function at $(x0, y0)$.

27. A graphics pipeline as recited in claim 23, wherein the interpolator processor is a fixed function single-instruction, multiple-data (SIMD) processor capable of operating on multiple data items with the same instruction.

28. A graphics pipeline as recited in claim 23, wherein the shading processor is an instruction-based computing element.